Theoretical Computer Science 107 (1993) 145–163
Elsevier

Quantifiers and approximation

Alessandro Panconesi and Desh Ranjan*

Department of Computer Science, Cornell University, Ithaca, NY 14853, USA

Abstract

Panconesi, A. and D. Ranjan, Quantifiers and approximation, Theoretical Computer Science 107 (1993) 145–163.

We investigate the relationship between logical expressibility of NP optimization problems and their approximation properties. First such attempt was made by Papadimitriou and Yannakakis (1988), who defined the class of NPO problems MAX NP. We show that many important optimization problems do not belong to MAX NP and that, in fact, there are problems in P which are not in MAX NP. The problems that we consider fit naturally in a new complexity class that we call MAX Π_1. We prove that several natural optimization problems are complete for MAX Π_1 under approximation-preserving reductions. All these complete problems are not approximable unless P = NP. This motivates the definition of subclasses of MAX Π_1 that only contain problems which are presumably easier with respect to approximation. In particular, the class that we call RMAX(2) contains approximable problems and problems like MAX CLIQUE that are not known to be nonapproximable. We prove that MAX CLIQUE and several other optimization problems are complete for RMAX(2). All the complete problems in RMAX(2) share the interesting property that they either are nonapproximable or are approximable to any degree of accuracy.

1. Introduction

The approximation of NP optimization (NPO) problems is an important area in the theory of algorithms [7, 13]. Although there is a wealth of results providing ingenious algorithms for the approximation of individual problems, and several isolated proofs of nonapproximability of others (assuming P ≠ NP), there is a lack of unifying theoretical framework and the reasons why a problem enjoys particular approximation properties are not clear [1, 2, 4, 7, 12, 14, 15].

In order to develop a theory for the approximation of NPO problems, one has to define subclasses of NPO, with problems in the same subclass having similar approximation properties. Defining these classes in terms of Turing machines presents a fundamental problem; changing something "computationally insignificant" like the

* Supported by NSF research grant CCR 8823053.

value of a single bit can have an enormous effect on the approximation properties of the computed function.

Krentel [11] develops a theory of NPO problems where the complexity of a problem is measured by the number of queries that any P^{SAT} machine, computing the optimization function for the problem, makes to its SAT oracle. The results are elegant but are not related to approximation; for example, MAX KNAPSACK and MIN TSP are both complete for the class of functions in P^{SAT}, but while the first is approximable in a very strong sense, the second is not approximable unless $P=NP$.

It is possible to define approximation classes within NPO in terms of Turing machines [4]. The results are interesting, but it seems doubtful that meaningful problems can be proven complete in these classes.

To avoid the problems which arise in the Turing machine model, Papadimitriou and Yannakakis [14] introduce an approach based on the logical characterization of NP given by Fagin [5]; this result states that NP is the set of languages that are the generalized spectrum of a second-order existential formula, ranging over finite structures. They use this characterization to define a natural class of NPO problems, which they call MAX NP. Roughly speaking, a problem in MAX NP has the property that the set of its feasible solutions can be described by a formula of the type $\exists \bar{y} \Phi(\bar{y}, S)$, where Φ is quantifier-free, S is a feasible solution and $\bar{y}$ ranges over the input structure, such as a graph or a boolean formula. Interestingly enough, they show that all the problems in MAX NP are approximable and that there is a *uniform* fashion in which they can be approximated.

This raises two questions. The first is: If there are approximable problems that are not in this class, that is, what is the expressive power of the class? The second, more general question, is: What is the relationship, if any, between the logical representation of a problem and its approximation properties?

In this paper, we first prove that the expressive power of MAX NP is rather limited. We prove that well-known and important problems like MAX CLIQUE, MAX 3DM, and MAX 3SP (optimization versions of 3DM and SET PACKING) are not in MAX NP. It is not known if MAX CLIQUE is approximable but we prove that MAX 3DM and MAX 3SP are approximable. In fact, we also prove the stronger fact that MAX NP does not even capture all of P because we show that the problem of finding a maximum matching in a graph cannot be expressed in MAX NP. These are expressibility results and do not rely on any assumptions (such as $P \neq NP$).

It turns out that all these problems which cannot be expressed as problems in MAX NP have a similar logical structure and they fit nicely into a new class that we call MAX Π_1. Loosely speaking, these problems have the property that the set of feasible solutions can be described by means of a first-order formula of the type $\forall \bar{y}\ \Phi(\bar{y}, S)$.

We investigate the structure of the class MAX Π_1 and find natural complete problems under reductions that preserve approximability [4, 12, 14, 15]. For example, we prove that, given a boolean formula, the problem of finding a satisfying assignment that sets to true the maximum number of variables (we call this problem MAX ONES) is

MAX Π_1-complete. MAX Π_1 in its full fledged form turns out to be too expressive; the complete problems for MAX Π_1 are not approximable unless P = NP. But the problems like MAX 3DM, MAX CLIQUE, and MAX 3SP do not seem to be that difficult and, in fact, they are either approximable or are not known to be nonapproximable. We define subclasses of MAX Π_1 which still capture these problems and where the approximability of the complete problems for the class is an open question. These subclasses are defined by restricting the structure of the logical formulae allowed to express the problems. The motivation for the constraints imposed comes from observing the similarity in the expressions for the problems mentioned above. The major restriction that is imposed corresponds to saying that if S is a feasible solution for the problem and $S' \subset S$, then so is S'. The smallest and most interesting of these subclasses contains MAX k-DM and MAX k-SP, and has MAX CLIQUE and MAX GRAPH k-COLORING as complete problems. The other classes have a natural generalization of MAX CLIQUE as their complete problems. All of the complete problems share the interesting property that either they are nonapproximable or are approximable within any fixed ratio.

This paper is organized as follows. Section 2 contains the necessary definitions. In Section 3, we prove that MAX CLIQUE, MAX 3DM, and MAX 3SP, and the maximum matching problem are not in MAX NP. We then introduce the class MAX Π_1 and prove that the above problems belong to it. In Section 4, we prove the MAX Π_1-completeness of the problems MAX ONES and MAX NSF with respect to approximation-preserving reductions (MAX NSF is the following problem: given a set of CNF formulae, find the maximum number of satisfiable ones). In Section 5, we define a subclass of MAX Π_1, the class RMAX, and prove completeness results for several optimization problems.

2. Definitions

Definition 2.1. An NPO problem is a tuple $F = (\mathscr{I}_F, S_F, f_F, opt)$, where

- $\mathscr{I}_F \subseteq \Sigma^*$ is the set of *input instances*. It is recognizable in polynomial time.
- $S_F(x)$ is the set of *feasible solutions* on input $x \in \mathscr{I}_F$. We require that $\forall x \in \mathscr{I}_F$, $S_F(x) = \{y \mid |y| \leqslant q_F(|x|) \wedge \pi_F(x, y)\}$, where q_F is a polynomial and π_F is a polynomial-time-computable predicate. q_F and π_F depend only on F.
- $f_F : \mathscr{I}_F \times \Sigma^* \to N$, the *objective function*, is a polynomial-time-computable function. $f_F(x, y)$ is defined only when $y \in S_F(x)$.
- $opt \in \{\max, \min\}$.

Solving an optimization problem F given the input $x \in \mathscr{I}_F$, means finding a $y \in S_F(x)$ such that $f_F(x, y)$ is optimum. The optimum value of F on input x is defined as

$$opt_F(x) = \underset{y \in S_F(x)}{opt} \; f_F(x, y).$$

As an example we briefly state how MAX CLIQUE can be expressed in the above formalism. The set of input instances is the set of all encodings of undirected graphs

$G=(V,E)$ over Σ^*. The set of feasible solutions for G is the set of all cliques contained in G. The objective function is the cardinality $\|C\|$ of a given clique. The goal is to find a clique of maximum cardinality. In this paper we consider only the maximization problems.

It seems possible to define NPO more concisely in the following way. We use $N(x,y)$ to indicate the final output of a nondeterministic Turing machine (NDTM) $N(x)$ along the computation path y. If we interpret $N(x,y)$ as a natural number then $F\in$NPO iff there exists a polynomial-time NDTM N such that, for all $x\in\mathscr{I}_F$,

$$opt_F(x)=\max_y N(x,y).$$

However, this definition is unsatisfactory as it does not explicitly state what the set of feasible solutions and the objective function are, and we believe that it is essential to separate these two objects if one wants to develop a theory of approximation.

To define approximation we need to define a notion of error [7, 9, 13].

Definition 2.2. The *relative error* of a feasible solution with respect to the optimum of an NPO problem F is defined as

$$\mathscr{E}_F(x,y)=\frac{|opt_F(x)-f_F(x,y)|}{opt_F(x)},$$

where $y\in S_F(x)$.

Definition 2.3. An NPO problem F is *ε-approximable* if there exists a polynomial-time algorithm A such that, for all instances x of F, (i) $A(x)\in S_F(x)$ and (ii) $\mathscr{E}_F(x,A(x))\leqslant\varepsilon$. A problem is *approximable* if there is an $\varepsilon\in(0,1)$ such that it is ε-approximable.

Note that the above definition is only meant for maximization problems. For minimization problems the definition would be the same except that ε could be any real greater than zero.

Definition 2.4. APX is the class of all approximable NPO problems.

Examples of problems in APX are MAX SAT, MAX CUT, MIN Δ TSP, MIN BIN PACKING and MIN NODE COVER [7, 13, 9].

It is well known that some NPO problems are ε-approximable for *any* $\varepsilon>0$ [7, 13].

Definition 2.5. An NPO problem is said to have a *polynomial-time approximation scheme* if there exists an algorithm $A(x,\varepsilon)$ such that, for all ε and all $x\in\mathscr{I}_F$, (i) $A(x,\varepsilon)\in S_F(x)$ and (ii) $\mathscr{E}_F(x,A(x,\varepsilon))\leqslant\varepsilon$. The complexity of A must be polynomial for any *fixed* ε.

To clarify the definition, the complexity of a polynomial-time approximation scheme can be something like $2^{1/\varepsilon}p(|x|)$ or something like $|x|^{1/\varepsilon}$; these cases actually arise in practice [8, 7].

Sometimes the dependence on ε is also polynomial.

Definition 2.6. An NPO problem is said to have a *fully polynomial-time approximation scheme* if it has a polynomial-time approximation scheme whose complexity is of the kind $p(1/\varepsilon, |x|)$, where p is a polynomial.

Definition 2.7. PTAS is the class of NPO problems that have a polynomial-time approximation scheme.

Definition 2.8. FPTAS is the class of NPO problems that have a fully polynomial-time approximation scheme.

Many scheduling problems are known to be in PTAS [8]. MAX KNAPSACK is an example of a problem belonging to the class FPTAS [13].

It follows directly from the definitions that FPTAS $\subseteq$ PTAS $\subseteq$ APX $\subseteq$ NPO. It is not difficult to see that these inclusions are strict if $P \neq NP$ [4, 7]. One of the most important open problems in the area is to understand under what conditions a problem is in APX or in PTAS. A variety of problems can be proved to be in APX but it is not known if they are in PTAS. Some examples are MAX SAT, MAX CUT, MIN Δ TSP, and MIN NODE COVER.

In a recent paper, Papadimitriou and Yannakakis [14] have addressed this question. Instead of defining NPO problems in terms of Turing machines, they use a logical characterization of NP due to Fagin [5]; a language is in NP if and only if it is the generalized spectrum of a second-order existential formula, ranging over finite structures.

For example, $\varphi \in \mathrm{SAT}$ if and only if

$$\exists T\, \forall c\, \exists x\ \ (P(x,c) \wedge T(x)) \vee (N(x,c) \wedge \neg T(x)).$$

Intuitively, T is a second-order variable that ranges over truth assignments; φ is described by means of the two binary predicates P and N; $P(x, c) = \mathrm{TRUE}$ iff variable x appears positive in clause c. Similarly, $N(x,c) = \mathrm{TRUE}$ iff variable x appears negated in clause c. The formula $(P(x,c) \wedge T(x)) \vee (N(x,c) \wedge \neg T(x))$ ensures that T sets to true at least one literal for each clause.

In general, for any language L in NP there is a *quantifier-free* formula Φ_L such that

$$I \in L \Leftrightarrow \exists S\, \forall \bar{x}\, \exists \bar{y}\ \ \Phi_L(I, S, \bar{x}, \bar{y})$$

(see [5, 10]). Informally, the instance I is described with a finite structure $I = \{A, P_1^{a_1}, \ldots, P_k^{a_k}\}$, where $P_i^{a_i} \subseteq A^{a_i}$, and A is a finite set. In the formula Φ_L, I stands for the set of predicates $P_i^{a_i}$ (this is an abuse of notation; for a more formal description see [5]). $S \subseteq A^s$ is a predicate of arity s describing the solution (e.g. a satisfying assignment), and $\bar{x}, \bar{y}$ are vectors of fixed arity of elements of A. We could consider a more general format where S too is a collection of predicates; in this paper, we consider the case where S is a single predicate for the sake of simplicity, but most of our proofs can be generalized.

It is important to realize that the formula Φ_L is the same for all instances I. In particular, it is of fixed size, and the arities of the vectors $\bar{x}, \bar{y}$, together with the arities of the predicates appearing in I and S, are fixed.

This formalism can be used to express NPO problems too. Again, for the sake of clarity, we consider an example. Take the problem MAX SAT: given a boolean formula φ in CNF, find an assignment that maximizes the number of clauses set to true.

Let $\Phi(x,c,P,N,T)$ be an abbreviation for $(P(x,c) \wedge T(x)) \vee (N(x,c) \wedge \neg T(x))$. Then for all instances φ the following holds

$$opt_{\text{MAX SAT}}(\varphi) = \max_{T} \| \{ c \mid \exists x \; \Phi(x,c,P,N,T) \} \|.$$

Definition 2.9 (Papadimitriou and Yannakakis [14]). MAX NP is the class of NPO problems F such that

$$opt_F(I) = \max_{S} \| \{ \bar{x} \mid \exists \bar{y} \Phi_F(I,S,\bar{x},\bar{y}) \} \|,$$

where Φ_F is a quantifier-free formula.

In our logical formulae we allow the use of the equality predicate with the usual interpretation.

Theorem 2.10 (Papadimitriou and Yannakakis [14]). *Every problem in* MAX NP *is* ε*-approximable for some* $\varepsilon \in (0,1)$, *i.e.* MAX NP $\subseteq$ APX.

To address the question of which MAX NP problems are in PTAS, Papadimitriou and Yannakakis [14] introduce a suitable approximation-preserving reducibility. We use a more general definition [4, 12].

Definition 2.11. Given two NPO problems F and G, a PTAS-*preserving reduction* (P-reduction) from F to G is a triple $f = (t_1, t_2, c)$ such that

(i) t_1, t_2 are polynomial-time-computable functions and $c:(0,1) \to (0,1)$;
(ii) $t_1 : \mathscr{I}_F \to \mathscr{I}_G$, and $t_2 : \mathscr{I}_F \times S_G(t_1(x)) \to S_F(x)$;
(iii) $\forall x \in \mathscr{I}_F$ and $\forall y \in S_G(t_1(x))$, if $\mathscr{E}_G(t_1(x), y \leqslant c(\varepsilon)$ then $\mathscr{E}_F(x, t_2(x,y)) \leqslant \varepsilon$.

Most of the reductions in this paper will actually be a much stronger form of this reduction.

Definition 2.12. A P-reduction from F to G is said to be an *approximation-preserving reduction* (A-reduction) if $c(\varepsilon) = \varepsilon$.

In a P-reduction, we use t_1 to map instances of F into instances of G, and t_2 to map an approximated solution for G back into an approximated solution of F. The relation among t_1, t_2 and c ensures that the following propositions hold.

Proposition 2.13. *If* $G \in \text{PTAS}$ *and* $F \leqslant_P G$, *then* $F \in \text{PTAS}$.

Proposition 2.14. P-*reductions compose, i.e. if* $F \leqslant_P G$ *and* $G \leqslant_P H$, *then* $F \leqslant_P H$.

Several natural problems are MAX NP-complete under P-reductions [14]. From Proposition 2.13, Proposition 2.15 follows.

Proposition 2.15. *If a* MAX NP-*hard problem is in* PTAS, *then* MAX NP $\subseteq$ PTAS.

In their paper, Papadimitriou and Yannakakis have left it as an open question if there are approximable optimization problems that are not in MAX NP. We answer this question positively in Section 3.

3. Expressiveness of MAX NP

In this section, we show that certain important optimization problems are not in MAX NP. In fact, we show that there are approximable problems and polynomially computable problems that are not in MAX NP. We first show that MAX CLIQUE is not in MAX NP; it is not known whether MAX CLIQUE is approximable. Then we introduce two large classes of matching and set packing problems. For these, we prove that they are approximable and that they do not belong to MAX NP. All these problems naturally belong to a new complexity class that we call MAX Π_1.

The following theorem is motivated by a more general principle,

$$\mathscr{A} \models \exists x\ \Phi(x) \wedge \mathscr{A} \subseteq \mathscr{B} \Rightarrow \mathscr{B} \models \exists x\ \Phi(x),$$

where $\Phi(x)$ is quantifier-free, and $\mathscr{A} \subseteq \mathscr{B}$ means that $\mathscr{A}$ is a submodel of $\mathscr{B}$ [3].

The proof of the following theorem is due to Dexter Kozen.

The statement and proof carry the implicit assumption that graphs are represented in the usual manner, i.e. as finite structures $G = (V, E)$, where V is the set of vertices and E is the edge predicate. We remark on other representations after the proof.

Theorem 3.1. MAX CLIQUE *is not in* MAX NP.

Proof. The proof is by contradiction. Assume that MAX CLIQUE belongs to MAX NP, meaning that for all graphs G,

$$opt_{\text{CLQ}}(G) = \max_S \|\{\bar{x} \mid \exists \bar{y}\ \Phi(\bar{x}, \bar{y}, E, S)\}\|,$$

where Φ is quantifier-free, and for some fixed $r, s, t > 0$, $\bar{x} = (x_1, \ldots, x_t)$, $\bar{y} = (y_1, \ldots, y_r)$, and $S \subseteq V^s$. Let us consider one particular $G_1 = (V_1, E_1)$ (with the only requirement that it has a nonempty edge set) and let S_1 be such that

$$opt_{\text{CLQ}}(G_1) = \|\{\bar{x} \mid \exists \bar{y}\ \Phi(\bar{x}, \bar{y}, E_1, S_1)\}\|.$$

Now we construct a new graph G_{new}. Let $G_2=(V_2,E_2)$ be an isomorphic copy of G_1 and let $G_{\text{new}}=(V_1\cup V_2,E_1\cup E_2)$ i.e. there are no edges between G_1 and G_2. Hence, $opt_{\text{CLQ}}(G_{\text{new}})=opt_{\text{CLQ}}(G_1)=opt_{\text{CLQ}}(G_2)\overset{\text{def}}{=}\text{OLDVALUE}$. We claim that

$$\max_S \|\{\bar{x}\mid\exists\bar{y}\ \Phi(\bar{x},\bar{y},E_{\text{new}},S)\}\|\geqslant 2\cdot\text{OLDVALUE}.$$

Given a tuple $\bar{x}$ of vertices in G_1 we indicate with $\bar{x}'$ the tuple made of the corresponding elements in G_2. Similarly, S_2 is the "isomorphic copy" of S_1; that is,

$$\langle a_1,\dots,a_s\rangle\in S_1 \;\Leftrightarrow\; \langle a_1',\dots,a_s'\rangle\in S_2.$$

We have that

$$\exists\bar{y}\ \Phi(\bar{a},\bar{y},E_1,S_1) \Leftrightarrow \exists\bar{y}\ \Phi(\bar{a}',\bar{y},E_2,S_2).$$

Choose $S_{\text{new}}=S_1\cup S_2$. We make the subclaim that, for all $\bar{a}\in V_1^t$,

$$\exists\bar{y}\ \Phi(\bar{a},\bar{y},E_1,S_1) \;\Rightarrow\; \exists\bar{y}\ \Phi(\bar{a},\bar{y},E_{\text{new}},S_{\text{new}}).$$

To see this, assume $\bar{a}$ to be such that $G_1,S_1\models\exists\bar{y}\ \Phi(\bar{a},\bar{y},E,S)$. This implies that there exists $\bar{b}$ such that $\Phi(\bar{a},\bar{b},E_1,S_1)$ is true. We show that $\Phi(\bar{a},\bar{b},E_{\text{new}},S_{\text{new}})$ is also true. We show that the truth values of the atoms of Φ in the two cases are the same. The atoms of $\Phi(\bar{a},\bar{b},E_{\text{new}},S_{\text{new}})$ are of the form $E_{\text{new}}(\bar{z})$, $S_{\text{new}}(\bar{w})$, where $\bar{z}$ and $\bar{w}$ are tuples of elements taken from the set $\{a_1,\dots,a_t,b_1,\dots,b_r\}$, or of the form $x=y$, where x and y range over $\{a_1,\dots,a_t,b_1,\dots,b_r\}$. But then, since $S_{\text{new}}=S_1\cup S_2$ and $E_{\text{new}}=E_1\cup E_2$.

$$E_{\text{new}}(\bar{w}) \;\Leftrightarrow\; E_1(\bar{w}) \quad\text{and}\quad S_{\text{new}}(\bar{z}) \;\Leftrightarrow\; S_1(\bar{z}).$$

A simple structural induction of formulae then shows that

$$\Phi(\bar{a},\bar{b},E_1,S_1) \;\Leftrightarrow\; \Phi(\bar{a},\bar{b},E_{\text{new}},S_{\text{new}}),$$

which proves the subclaim. Similarly, $\exists\bar{y}\ \Phi(\bar{x}',\bar{y},E_2,S_2)\Rightarrow\exists\bar{y}\ \Phi(\bar{x}',\bar{y},E_{\text{new}},S_{\text{new}})$, and, hence,

$$\|\{\bar{x}\mid\exists\bar{y}\ \Phi(\bar{x},\bar{y},E_{\text{new}},S_{\text{new}})\}\|\geqslant 2\cdot\text{OLDVALUE}$$

because $\{\bar{x}\mid\exists\bar{y}\ \Phi(\bar{x},\bar{y},E_1,S_1)\}$ and $\{\bar{x}'\mid\exists\bar{y}\ \Phi(\bar{x}',\bar{y},E_2,S_2)\}$ are disjoint. □

The theorem was proved under the assumption that a graph is a finite structure of the kind $G=(V,E)$. However, what we actually used in the proof were the following assumptions on the coding of graphs via finite structures. First, isomorphic graphs are represented by isomorphic structures and isomorphic structures represent isomorphic graphs. Second, if $G_1=(V_1,E_1)$ has a coding $G_1=(A_1,P_1^1,\dots,P_m^1)$ and $G_2=(V_2,E_2)$ has a coding $G_2=(A_2,P_1^2,\dots,P_m^2)$, then $G=(V_1\cup V_2,E_1\cup E_2)$ has a coding isomorphic to $G=(A_1\cup A_2,P_1^1\cup P_1^2,\dots,P_n^1\cup P_n^2)$. These conditions are satisfied by any reasonable encoding of graphs.

We now introduce a family of optimization problems. We first show that they are approximable and then that they do not belong to MAX NP.

This family is a natural generalization of the MAXIMUM MATCHING problem.

Suppose we are given a set of k-tuples $T=\{T_1,\ldots,T_n\}\subseteq A_1\times A_2\times\cdots\times A_k$, where the A_i's are pairwise disjoint sets. Say, two tuples are *compatible* if they differ in all k components. Then a set $M\subseteq T$ is a *matching* if every two k-tuples in M are compatible.

MAX k-DIMENSIONAL MATCHING (MAX k-DM)

Instance: A collection of k-tuples $T=\{T_1,\ldots,T_n\}$.

Problem: Find the maximum size matching.

When $k=2$, MAX k-DM is equivalent to the MAXIMUM MATCHING problem on bipartite graphs that is known to be in P. MAX 3DM is the optimization version of the NP-complete problem 3DM [6].

Proposition 3.2. *For all* $k\geqslant 2$, MAX k-DM *is in* APX.

Proof. One can show that the size of any *maximal* matching is at least $1/k$ of the size of a maximum matching. □

The next theorem shows that MAX NP does not include all polynomially computable optimization problems.

Theorem 3.3. MAX 2DM *is not in* MAX NP.

Proof. The proof is similar to that of Theorem 3.1. Assume, for contradiction, that MAX 2DM$\in$MAX NP. Then there exists a formula Φ such that for all instances I of MAX 2DM,

$$opt_{2\text{DM}}(I)=\max_S \|\{\bar{x}\mid\exists\bar{y}\ \Phi(\bar{x},\bar{y},I,S)\}\|.$$

Consider an instance $I_1=\{T_1,\ldots,T_n\}$ such that $opt_{2\text{DM}}(I_1)=n$, i.e. I_1 is a set of n pairwise compatible pairs (we can look at I_1 as a collection of n disjoint edges).

From our assumption for contradiction, we have that there is S_1 such that

$$opt_{2\text{DM}}(I_1)=\|\{\bar{x}\mid\exists\bar{y}\ \Phi(\bar{x},\bar{y},I_1,S_1)\}\|=n.$$

Let $\bar{x}_1,\ldots,\bar{x}_n$ be the tuples satisfying the above formula. Consider $\bar{x}_1$ and suppose, without loss of generality, that it contains a_1, i.e. $\bar{x}_1=(a_1,u_2,\ldots,u_k)$, and that $T_1=(a_1,b_1)$.

We now construct another instance I_2 by simply replacing a_1 with a brand new element a_0. Let $I_2=\{T_0,T_2,\ldots,T_n\}$, where $T_0=(a_0,b_1)$. I_2 is made of the same tuples of I_1 except the first, T_0. T_0 and T_1 only differ for the first component, namely a_1. We choose a_0 so that I_2 is made of n mutually compatible tuples. Now define S_2 to be the same set as S_1, provided any occurrence of a_1 is replaced by an occurrence of a_0, and define $\bar{z}_i$ to be the same tuple as $\bar{x}_i$, provided the same substitution takes place. Then

$$\|\{\bar{z} \mid \exists \bar{y}\ \Phi(\bar{z}, \bar{y}, I_2, S_2)\}\| = n.$$

If we now consider the new instance $I_{\text{new}} = I_1 \cup I_2$ and define $S_{\text{new}} = S_1 \cup S_2$, we have that $opt_{2\text{DM}}(I_{\text{new}}) = n$ but

$$\max_S \|\{\bar{w} \mid \exists \bar{y}\ \Phi(\bar{w}, \bar{y}, I_{\text{new}}, S)\}\| \geqslant \|\{\bar{w} \mid \exists \bar{y}\ \Phi(\bar{w}, \bar{y}, I_{\text{new}}, S_{\text{new}})\}\| \geqslant n+1$$

because

$$\|\{\bar{x}_1, \ldots, \bar{x}_n\} \cup \{\bar{z}_1, \ldots, \bar{z}_n\}\| \geqslant n+1.$$

This contradiction shows that MAX 2DM $\notin$ MAX NP. □

Basically, the same proof applies to MAX k-DM, for all $k \geqslant 2$.

Corollary 3.4. *For all* $k \geqslant 2$, MAX k-DM *does not belong to* MAX NP.

We now introduce another family of problems, similar to MAX k-DM. Given a collection of sets of cardinality k, $S = \{S_1, \ldots, S_n\}$, we define a *packing* to be a collection of pairwise disjoint sets: $S_i \in C \wedge S_j \in C \Rightarrow S_i \cap S_j = \emptyset$.

MAX k-SET PACKING (MAX k-SP)

Instance: A collection $S = \{S_1, \ldots, S_n\}$ of sets, where each S_i has cardinality k.

Problem: Find a packing of maximum size.

MAX k-SP is the natural optimization version of the problem SET PACKING [7]. We claim, without proof, that the following theorems hold. Their proofs are very similar to the theorems we saw for MAX k-DM.

Theorem 3.5. *For all* $k \geqslant 2$, MAX k-SP *is in* APX.

Theorem 3.6. *For all* $k \geqslant 2$, MAX k-SP *does not belong to* MAX NP.

All the problems we introduced in this section fit nicely in a new complexity class.

Definition 3.7. MAX Π_1 is the class of NP optimization problems F such that, for all input instances I,

$$opt_F(I) = \max_S \|\{\bar{x} \mid \forall \bar{y}\ \Phi(I, S, \bar{x}, \bar{y})\}\|.$$

As an example, consider MAX CLIQUE. It is easy to see that, for all graphs G,

$$opt_{\text{CLQ}}(G) = \max_C \|\{x \mid C(x) \wedge \forall yz(C(y) \wedge C(z) \rightarrow E(y,z) \vee y=z)\}\|,$$

where x, y and z range over vertices and $E(y,z) = \text{TRUE}$ iff $(y,z) \in E$.

The proposition which we state next has a trivial proof, which is omitted.

Proposition 3.8. MAX CLIQUE, MAX k-DM, *and* MAX k-SP *belong to* MAX Π_1.

MAX Π_1 is a natural way of expressing many NPO problems. In Section 4 we prove completeness for natural variants of SAT.

In particular, our canonical complete problem is the following.

MAX NUMBER OF ONES (MAX ONES)

Instance: A boolean formula φ in 3CNF.

Problem: Find a satisfying assignment with the maximum number of variables set to TRUE.

We can express MAX ONES as a MAX Π_1 problem as follows. As in the case of 3SAT the instance is coded by means of four predicates $C_0, \ldots, C_3$, where $C_i(x, y, z) = \text{TRUE}$ iff φ has a clause whose variables are x, y, and z and where the first i among its variables appear negated (e.g. $C_2(x, y, z)$ means $(\bar{x} \vee \bar{y} \vee z)$ is a clause) [14]. Then

$$opt_{\text{ONES}}(\varphi) = \max_T \|\{x \mid T(x) \wedge \forall yzw\ \Phi(\varphi, T, x, y, z, w)\}\|,$$

where

$$\begin{aligned}\Phi(\varphi, T, x, y, z, w) = {} & (C_0(y, z, w) \rightarrow T(y) \vee T(z) \vee T(w)) \\ & \wedge (C_1(y, z, w) \rightarrow \neg T(y) \vee T(z) \vee T(w)) \\ & \wedge (C_2(y, z, w) \rightarrow \neg T(y) \vee \neg T(z) \vee T(w)) \\ & \wedge (C_3(y, z, w) \rightarrow \neg T(y) \vee \neg T(z) \vee \neg T(w)).\end{aligned}$$

4. Structural properties of MAX Π_1

In this section we exhibit complete problems for the class MAX Π_1. We also show that the complete problems for the class are nonapproximable unless P = NP. Our first MAX Π_1-complete problem is the following.

MAX NUMBER OF SATISFIABLE FORMULAE (MAX NSF)

Instance: A set of 3CNF formulae $\{\varphi_1, \ldots, \varphi_n\}$.

Problem: Find a truth assignment to the variables such that the maximum number of the formulae are satisfied.

In this problem, the set of feasible solutions of nonzero weight are the assignments satisfying at least one formula φ_i; this implies that approximating MAX NSF is NP-hard.

Theorem 4.1. MAX NSF *is* MAX Π_1*-complete under A-reductions.*

Proof. We first show that MAX NSF $\in$ MAX Π_1. Informally, this is because the optimum value on instance I can be expressed as

$$opt_{\text{MAX NSF}}(I) = \max_T \|\{i \mid \varphi_i(T) = \text{TRUE},\ 1 \leqslant i \leqslant n\}\|,$$

where I is the input instance $\{\varphi_1, \varphi_2, \ldots, \varphi_n\}$ and T is a unary predicate which is basically a truth assignment to the variables in $\{\varphi_1, \varphi_2, \ldots, \varphi_n\}$.

To write this formally, we may suppose that I is presented via two 3-ary predicates P and N, where $P(i,j,k)$ is true iff variable x_k occurs positively in the jth clause of the formula φ_i, C_{ij}, and $N(i,j,k)$ is true iff variable x_k occurs negatively in C_{ij}. Then, more precisely,

$$opt_{\text{MAX NSF}}(I)=\max_T \|\{i \mid \forall j k_1 k_2 k_3\ \Phi(N,T,i,j,k_1,k_2,k_3)\}\|,$$

where

$$\begin{aligned}\Phi=&(P(i,j,k_1)\wedge P(i,j,k_2)\wedge P(i,j,k_3)\rightarrow T(k_1)\vee T(k_2)\vee T(k_3))\\ &\wedge(P(i,j,k_1)\wedge P(i,j,k_2)\wedge N(i,j,k_3)\rightarrow T(k_1)\vee T(k_2)\vee\neg T(k_3))\\ &\wedge(P(i,j,k_1)\wedge N(i,j,k_2)\wedge N(i,j,k_3)\rightarrow T(k_1)\vee\neg T(k_2)\vee\neg T(k_3))\\ &\wedge(N(i,j,k_1)\wedge N(i,j,k_2)\wedge N(i,j,k_3)\rightarrow\neg T(k_1)\vee\neg T(k_2\vee\neg T(k_3)).\end{aligned}$$

Second, we establish the completeness of MAX NSF.

Let F be any optimization problem in MAX Π_1, and let f_F be its optimization function. Then

$$opt_F(I)=\max_S \|\{\bar{x} \mid \forall\bar{y}\ \Psi(\bar{x},\bar{y},I,S)\}\|.$$

Recall that $\bar{x},\bar{y}$ represent fixed-arity tuples of variables. Hence, each tuple ranges over a polynomially sized domain (in the size of I). Let us enumerate the domain of x as $\bar{a}_1,\bar{a}_2,\ldots,\bar{a}_m$ and the domain of y as $\bar{b}_1,\bar{b}_2,\ldots,\bar{b}_p$. Each $\bar{a}_i$ is a tuple of names for elements of the domain, which can be substituted for the respective variables of $\bar{x}$ in Ψ; similarly, the names $\bar{b}_j$ can be substituted for $\bar{y}$. Then for each i, $1\leqslant i\leqslant m$, define φ_i to be the formula $\bigwedge_{1\leqslant j\leqslant p}\Psi(\bar{a}_i,\bar{b}_j,I,S)$. Each φ_i is a polynomially sized boolean formula whose variables are $S(v_1,\ldots,v_l)$, where S is an l-ary predicate. Moreover, there are exactly m of these formulae. Since the formula Ψ is fixed in terms of F, the time taken to put Ψ itself into CNF is immaterial. Then, with the introduction of new variables, Ψ can be changed into a 3CNF formula maintaining satisfiability. Hence, we can assume that each φ_i is a 3CNF formula.

Now observe that, for predicate assignment S_0 to S, the corresponding truth assignment S' to $\{S(v_1,\ldots,v_l) \mid (v_1,\ldots,v_l)\in domain(S)\}$ given by

$$S(v_1,\ldots,v_l)=\text{TRUE} \Leftrightarrow (v_1,\ldots,v_l)\in S_0$$

makes k-many formulae φ_i true iff $f_F(S_0)=k$. Hence, this is an A-reduction, which concludes the proof. □

We now show the MAX Π_1-completeness of MAX ONES with respect to P-reductions. We have already shown at the end of Section 3 that MAX ONES is in MAX Π_1. To show hardness, we first exhibit a reduction from MAX NSF into an intermediate problem, MAX DONES, and then reduce MAX DONES to MAX ONES. MAX DONES is the following problem.

MAX DISTINGUISHED ONES (MAX DONES)

Instance: A boolean formula $\varphi(X,Z)$, where $X=\{x_1,\ldots,x_n\}$ and $Z=\{z_1,\ldots,z_m\}$. The z_i's are the *distinguished* variables.

Problem: Find a satisfying assignment for φ with the maximum number of distinguished variables set to TRUE.

Lemma 4.2. MAX DONES *is* $\text{MAX}\,\Pi_1$*-complete with respect to A-reductions.*

Proof. MAX DONES can be written down as a $\text{MAX}\,\Pi_1$ problem in essentially the same way we wrote MAX ONES; besides the predicates C_i we need a predicate $D(z)$ that is TRUE iff z is a distinguished variable.

We now reduce MAX NSF to MAX DONES. Given an instance $\psi=\{\varphi_1(X),\ldots,\varphi_n(X)\}$ of MAX NSF we construct the formula

$$F(X,Z)=(\varphi_1 \vee \neg z_1) \wedge \cdots \wedge (\varphi_n \vee \neg z_n).$$

By distributing the z_i's over the clauses of φ_i we can see that F is a 4CNF formula that satisfies the following: there is an assignment that makes k formulae $\varphi_{i_1},\ldots,\varphi_{i_k}$ in ψ true if and only if there is a satisfying assignment for $F(X,Z)$ that sets $z_{i_1},\ldots,z_{i_k}$ to TRUE.

This is an A-reduction. To complete the proof we have to transform $F(X,Z)$ into a 3CNF formula. This can be done by introducing extra undistinguished variables y_i's; a clause $(x_1 \vee x_2 \vee x_3 \vee x_4)$ is mapped into the two clauses $(x_1 \vee x_2 \vee y_1) \wedge (\neg y_1 \vee x_3 \vee x_4)$. Since the y's are nondistinguished, this is again an A-reduction. □

Theorem 4.3. MAX ONES *is* $\text{MAX}\,\Pi_1$*-complete with respect to P-reductions.*

Proof. We have already established that MAX ONES $\in \text{MAX}\,\Pi_1$ at the end of the preceding section. To prove completeness, we transform MAX DONES into MAX ONES. Let $\varphi(X,Z)$, with $Z=\{z_1,\ldots,z_p\}$ and $X=\{x_1,\ldots,x_q\}$, be an instance of MAX DONES; we transform it into an instance $\psi(X,Y,Z,Z')$ of MAX ONES. In what follows, we will indicate with $\tau': X\cup Y\cup Z\cup Z'\to\{0,1\}$ a satisfying assignment for ψ, and with τ the restriction to $X\cup Z$ of τ'. The reduction we are going to show is such that $\psi(\tau'(X),\tau'(Y),\tau'(Z),\tau'(Z'))=\text{TRUE} \Leftrightarrow \varphi(\tau(X,Z))=\varphi(\tau'(X),\tau'(Z))=\text{TRUE}$.

The instance of MAX ONES is the following formula:

$$\psi(X,Y,Z,Z')=\varphi(X,Z) \wedge \beta(Z,Z') \wedge \alpha(X,Y,Z),$$

where Y and Z' are sets of brand new variables while X and Z are the same variables appearing in φ.

In ψ any true variable contributes to the weight of a satisfying assignment τ'. We would like the contribution of the x's and y's to be negligible with respect to that of the

z's. The mission of the subformula $\beta(Z,Z')$ is to amplify the weight carried by the variables z's. We define

$$\beta(Z,Z')=\bigwedge_{1\leqslant i\leqslant p}\left(\bigwedge_{1\leqslant j\leqslant 2l-1} z_i\leftrightarrow z_{ij}\right),$$

where $Z=\{z_1,\ldots,z_p\}$ and $Z'=\{z_{ij}\,|\,1\leqslant i\leqslant p,\ 1\leqslant j\leqslant 2l-1\}$. The index l is set to $l=q+r$, where r is the number of y's in ψ. l is selected after the construction of α is done. What β does is equivalent to assigning a weight of $2l$ to each z_i. Note that β can be expressed in CNF with clauses of two literals. Also note that any satisfying assignment for $\varphi(X,Z)$ automatically determines a satisfying assignment for $\beta(Z,Z')$ and that β can be expressed in CNF using $4lp$-many clauses of two literals each.

The mission of $\alpha(X,Y,Z)$ is to forbid truth assignments of ψ where some of the x_i's are set to true and all the z_i's are set to false. If this happened, we would have a solution of ψ with cost greater than zero mapped into a solution of φ of cost zero, i.e. approximated solutions would not be mapped into approximated solutions (recall that our transformation simply considers restrictions $\tau(X,Z)=\tau'(X,Z)$, where τ' satisfies ψ).

A way of implementing α would be to write down

$$\alpha=\bigwedge_{1\leqslant i\leqslant q}\left(\neg x_i\vee\bigvee_{1\leqslant j\leqslant q} z_j\right).$$

But these are clauses of unbounded length. To have clauses of length at most three, we transform each clause $(\neg x_i\vee z_1\vee\cdots\vee z_q)$ into

$$(\neg x_i\vee z_1\vee y_1)\wedge(\neg y_1\vee z_2\vee y_2)\wedge\cdots\wedge(\neg y_{q-1}\vee z_q).$$

It can be checked that $\alpha(X,Y,Z)$ so defined satisfies the two properties: (i) the truth of any y_i or x_i implies the truth of some z_j; (ii) any truth assignment τ' of $\varphi(Z,X)$ of nonzero cost can be extended to a truth assignment $\tau\supseteq\tau'$ satisfying $\alpha(X,Y,Z)$.

To summarize, $\psi(X,Y,Z,Z')$ can be expressed in 3CNF and the restrictions to $X\cup Z$ of its satisfying assignments form the set of nonzero cost satisfying assignments for $\varphi(X,Z)$.

We now have to show that the reduction is a P-reduction. The transformation can certainly be carried over in polynomial time.

Let $opt_{\mathrm{DONES}}(\varphi)=k$. By construction, it follows that $opt_{\mathrm{ONES}}(\psi)\geqslant 2lk$. It also follows that the possible weights for a solution τ' of ψ are $w(\tau')=0,2l+n_1,\ldots,2li+n_i,\ldots,2lk+n_k$, where $1\leqslant i\leqslant k$ and $n_i\leqslant l=q+r$ for all i's. Moreover, the relationship between a solution τ' and its restriction τ is given by $w(\tau')=2li+n_i\Rightarrow w(\tau)=i$ and $w(\tau')=0\Rightarrow w(\tau)=0$.

We want to show that

$$\frac{opt(\psi)-w(\tau')}{opt(\psi)}\leqslant\frac{\varepsilon}{2}\Rightarrow\frac{opt(\varphi)-w(\tau)}{opt(\varphi)}\leqslant\varepsilon.$$

In order to do so, it is enough to prove that

$$\frac{opt(\psi)-w(\tau')}{opt(\psi)} \geqslant \frac{opt(\varphi)-w(\tau)}{2\,opt(\varphi)}.$$

Consider a solution τ' of ψ. When $w(\tau')=0$ or $w(\tau')\geqslant 2lk$ the above equation holds. Suppose then that $w(\tau')=2li+n_i$ with $1\leqslant i\leqslant k-1$. We have

$$\frac{opt(\psi)-w(\tau')}{opt(\psi)} \geqslant \frac{2lk-w(\tau')}{2lk} \tag{1}$$

$$\geqslant \frac{2lk-l(2i+1)}{2lk} \tag{2}$$

$$= \frac{2k-(2i+1)}{2k}$$

$$\geqslant \frac{k-i}{2k}$$

$$= \frac{opt(\varphi)-w(\tau)}{2\,opt(\varphi)}.$$

Equation (1) holds since $opt(\psi)\geqslant 2lk$, while Eq. (2) holds since $w(\tau')=2li+n_i\leqslant l(2i+1)$. This concludes the proof. $\square$

The complete problems we saw are not approximable unless P = NP. However, we know that MAX Π_1 contains approximable problems, like MAX k-DM, and problems that are believed not to be approximable like MAX CLIQUE, and which have the interesting property that they are either not approximable or are in PTAS. It would be interesting to characterize, within MAX Π_1, classes whose problems share similar approximation properties. In Section 5, we see how it is possible to describe problems like MAX CLIQUE, MAX k-SP, and MAX k-DM by posing syntactic restrictions on the formulae Φ_F certifying membership of F in MAX Π_1.

5. Expressive power of restrictions of MAX Π_1

In Section 4, we saw that MAX Π_1 in its full generality has problems which are too hard for approximation. On the other hand, let us examine the expressions for the optimization functions for various problems we have been discussing, and which we proved are not in MAX NP:

- MAX CLIQUE. We have that $opt_{\mathrm{CLQ}}(G)=\max_C \| \{x \mid C(x) \wedge \forall uv\ \Phi(C,E,u,v\} \|$, where $\Phi(C,E,u,v)=(\neg C(u) \vee \neg C(v) \vee E(u,v))$.
- MAX 3DM. We have that $opt_{\mathrm{3DM}}(I)=\max_M \| \{\bar{a} \mid M(\bar{a}) \wedge \forall \bar{b}\bar{c}\ \Phi(M,T,\bar{b},\bar{c})\} \|$, where $\Phi(M,T,\bar{b},\bar{c})=[\neg M(\bar{b}) \vee T(\bar{b})] \wedge [\neg M(\bar{b}) \vee \neg M(\bar{c}) \vee \wedge_{i=1,2,3}(b_i\neq c_i)]$.

Here $\bar{a}$ stands for (a_1, a_2, a_3) and I stands for the input instance (A, T), where $T \subseteq A^3$ with $T(\bar{a}) = \text{TRUE}$ iff "*$\bar{a}$ is a triple*".

These problems are not only in MAX Π_1, but the fashion in which they are expressed is also rather similar. More precisely, all these problems can be expressed as

$$opt_F(I) = \max_S \{ \|S\| : \forall \bar{y}\ \Phi_F(\bar{y}, I, S) \},$$

where $\|S\|$ denotes $\|\{\bar{x}: S(\bar{x})\}\|$, $\bar{y}$ is a first-order variable and Φ_F is quantifier-free. Most importantly, *if Φ_F is expressed in CNF then all occurrences of S occur negatively.*

Definition 5.1. A problem $F \in \text{RMAX}(k)$ if its optimization function can be expressed as

$$opt_F(I) = \max_S \{ \|S\| : \forall \bar{y}\ \Phi(\bar{y}, I, S) \},$$

where Φ is a quantifier-free CNF formula with all occurrences of S in Φ being negative, S a single predicate appearing at most k times in each clause, and $\|S\|$ denotes $\|\{x: S(\bar{x})\}\|$.

Definition 5.2. $\text{RMAX} = \bigcup_k \text{RMAX}(k)$.

This subclass may seem very restricted in the beginning, but it captures many of the problems in MAX Π_1 which are provably not in MAX NP. In fact, most of the problems we have considered are in RMAX(2). Other problems which fall into this class include:

- MAX SP: This is a generalization of MAX k-SP. Given a collection $S_1, \ldots, S_n$ of finite sets, find a packing of maximum size. Note, MAX SP $= \bigcup_k$ MAX k-SP. This problem and MAX k-SP are in RMAX(2). Similarly, MAX DM and MAX k-DM are in RMAX(2).
- MAX INDEPENDENT SET: Given a graph, find the size of the maximum independent set. This problem is in RMAX(2).
- MAX GRAPH k-COLORING: Given a graph $G = (V, E)$ and an integer k, find the maximum number of vertices of G that can be colored with k colors such that no two adjacent vertices have the same color. This problem is in RMAX(2).
- MAX k-ANLSAT: This is the restriction of MAX ONES where all the variables in the input formula appear only negatively, and where every clause has at most k literals. This problem is in RMAX(k).
- MAX k-HYPERCLIQUE: An input instance is a k-hypergraph $H = (A, E)$, where A is a set and $E \subset \mathscr{P}(A)$ and $e \in E \Rightarrow 1 \leqslant |e| \leqslant k$. An element of E is called a hyperedge. A feasible solution is any set $W \subset A$ satisfying $\{u_1, \ldots, u_i\} \subseteq W \Rightarrow \{u_1, \ldots, u_i\} \in E$, $i \leqslant k$. Such a set is called a *k-hyperclique*. The goal is to find a k-hyperclique of maximum size. This problem is a generalization of the CLIQUE problem for graphs to hypergraphs and it is a trivial fact that MAX CLIQUE $\equiv_A$ MAX 2-HYPERCLIQUE.

Thus, there is a large class of problems which are in RMAX(k). We now establish two theorems. The first theorem shows that MAX k-ANLSAT is complete in RMAX(k)

with respect to A-reductions. The second theorem is about the equivalence of the families of problems MAX k-HYPERCLIQUE and MAX k-ANLSAT.

Theorem 5.3. *If* $F \in \mathrm{RMAX}(k)$ *then* $F \leqslant_{\mathrm{A}}$ MAX k-ANLSAT.

Proof. If $F \in \mathrm{RMAX}(k)$, it can be expressed with a formula Φ_F with k occurrences of the predicate S per clause and with all negative occurrences of S. The rest of the proof is similar to that of Theorem 4.1. □

Theorem 5.4. *For all* $k \geqslant 2$, MAX k-HYPERCLIQUE $\equiv_{\mathrm{A}}$ MAX k-ANLSAT.

Proof. We first show that MAX k-HYPERCLIQUE $\leqslant_{\mathrm{A}}$ MAX k-ANLSAT. Let $H=(A,E)$ be any k-hypergraph. Construct ϕ_H as follows. The variables of ϕ_H are $\{x_i \mid i \in A\}$. ϕ_H is a conjunction of all the clauses of the form $(\neg x_{i_1} \vee \neg x_{i_2} \vee \cdots \vee \neg x_{i_k})$, where $x_{i_j} \in var(\phi_H)$ and $\{i_1, i_2, \ldots, i_k\}$ is not a hyperedge in E. The literals may be repeated within a clause, in which case it is simplified. These are the only clauses of ϕ_H. It can now be checked that ϕ_H can be satisfied with $x_{i_1}, x_{i_2}, \ldots, x_{i_l}$ all set to TRUE if and only if $\{i_1, i_2, \ldots, i_l\}$ form a hyperclique in H.

To prove that MAX k-ANLSAT $\leqslant_{\mathrm{A}}$ MAX k-HYPERCLIQUE, use the inverse mapping. □

The last two theorems have interesting consequences.

Theorem 5.5. *The problems* MAX CLIQUE, MAX GRAPH k-COLORING, MAX SP, *and* MAX DM *are* RMAX(2)*-complete with respect to A-reductions.*

Proof. The completeness of MAX CLIQUE follows from Theorem 5.3 and the trivial fact that MAX CLIQUE $\equiv_{\mathrm{A}}$ MAX 2-HYPERCLIQUE. The remaining reductions are easy to obtain; for example, MAX CLIQUE $\leqslant_{\mathrm{A}}$ MAX GRAPH k-COLORING because an independent set can always be 1-colored. □

All the RMAX(2)-complete problems share a very interesting property: either they are not approximable or they are in PTAS. The reason why, for example, MAX CLIQUE shows this behavior is that given any graph G we can, in polynomial time, construct G' such that: (i) $opt(G')=opt(G)^2$; (ii) if $C' \subseteq G'$ is a clique of k vertices then we can find in polynomial time a clique $C \subset G$ of at least $\sqrt{k}$ vertices. This implies that if $|C'|/opt(x') \geqslant 1-\varepsilon$ then $|C|/opt(x) \geqslant \sqrt{1-\varepsilon}$. Since $\lim_{n\to\infty}(1-\varepsilon)^{1/2^n}=1$, we can obtain in G any approximation we want by iterating the above construction [6, 13].

The following definition generalizes this kind of situation. Recall that f_F is the objective function of the NPO problem F.

Definition 5.6. A problem $F \in \mathrm{NPO}$ is *self-improvable* if there is a P-reduction $r=(t_1, t_2, c)$ from F to itself and a function h such that

(i) $h:(0,1)\to(0,1)$, h is monotone increasing, and $\lim_{n\to\infty} h^n(z)=1$.

(ii) $$\frac{f_F(x,y)}{opt_F(x)} \geqslant h\left(\frac{f_F(x',y')}{opt_F(x')}\right),$$

where $x'=t_1(x)$, $y'\in S_F(x')$, and $y=t_2(x,y')$.

If a problem is self-improvable, then it is either in NPO − APX or in PTAS. The reason is that we can apply the reduction n times to map x into $t_1^n(x)$; an error of ε in the solution of $t_1^n(x)$ corresponds to an error ε_n in the solution of x, where ε_n tends to 0 as n tends to infinity. This can be seen as follows: let $x_n=t_1^n(x_0)$ and $y_n\in S_F(x_n)$; from the above definition it follows that

$$\frac{f_F(x_0,y_0)}{opt_F(x_0)} \geqslant h^n\left(\frac{f_F(x_n,y_n)}{opt_F(x_n)}\right),$$

where y_0 is obtained by repeated applications of t_2 on appropriate arguments starting from x_{n-1} and y_n.

For example, MAXCLIQUE is self-improvable with $h(z)=z^{1/2}$ [6, 13].

Fact 5.7. *If F is A-equivalent to G and G is self-improvable, so is F.*

We then have the following corollary.

Corollary 5.8. *All the A-complete problems of Theorem* 5.5 *are self-improvable.*

Note that these results are obtained without directly mapping these problems to themselves.

6. Conclusion

We have investigated the relationship between the logical expressibility of NPO problems and their approximation properties. To summarize, we have first shown that class MAX NP is rather weak in its expressive power. We have then defined another class of NPO problems based on logical structure. For this class we have demonstrated complete problems; moreover, we have obtained interesting subclasses where the complete problems have similar properties with respect to approximation and, in addition, they all have the property of self-improvability. This work is a step in the direction of developing a general framework for establishing a connection between the logical structure of a problem and its approximation properties and we hope that it provides an impetus for the same.

Acknowledgment

We thank Dexter Kozen for proving Theorem 3.1. We are indebted to Radhakrishnan Jagadeesan, Tushar Deepak Chandra and Steve Vavasis for insightful criticism of the ideas presented in this paper. We thank Vijay Vazirani and David Williamson for their comments on an earlier draft of this paper. The first author thanks Pierluigi Crescenzi and Riccardo Silvestri; discussions with them were the starting point of this work. The second author gratefully acknowledges the support of his advisor Juris Hartmanis. Finally, we thank the unknown referees for suggesting several improvements, both to the content of the paper and the style of our presentation.

Note added in proof

In a recent breakthrough, Arora and Safra [16] have proved that MAX CLIQUE is not approximable unless $P=NP$, thus solving a long-standing open question. The result obviously extends to other RMAX(2)-complete problems. Also, in another paper Arora et al. [17] showed that MAX SNP-hard problems do not have a polynomial-time approximation scheme unless $P=NP$. In particular, this means that MAX k-DM and MAX k-SP$\notin$PTAS unless $P=NP$.

References

[1] G. Ausiello, A. Marchetti Spaccamela and M. Protasi, Toward a unified approach for the classification of NP-complete problems, *Theoret. Comput. Sci.* **12** (1980) 83–96.

[2] D. Bruschi, D. Joseph and P. Young, A structural overview of NP optimization problems, Computer Sci. Tech. Report 861, Univ. of Wisconsin, Madison, 1989.

[3] C.C. Chang and H.J. Keisler, *Model Theory* (North-Holland, Amsterdam, 1973).

[4] P. Crescenzi and A. Panconesi, Completeness in approximation classes, *Inform. and Comput.* **93** (1991) 241–262.

[5] R. Fagin, Generalized first-order spectra, and polynomial-time recognizable sets, in: R. Karp, ed., *Complexity and Computations* (Amer. Mathematical Soc., Providence, RI, 1974).

[6] M.R. Garey and D. Johnson, The complexity of near-optimal graph coloring, *J. ACM* **23** (1976) 43–49.

[7] M.R. Garey and D. Johnson, *Computers and Intractability: A Guide to the Theory of* NP-*Completeness* (Freeman, San Francisco, 1979).

[8] D. Hochbaum and D. Shmoys, Using dual approximation algorithms for scheduling problems: theoretical and practical results, *J. ACM* **34** (1987) 144–162.

[9] D. Johnson, Approximation algorithms for combinatorial problems, *J. Comput. System Sci.* **9** (1974) 256–278.

[10] P.G. Kolaitis, private communication.

[11] M.W. Krentel, The complexity of optimization problems, *J. Comput. System Sci.* **36** (1988) 490–509.

[12] P. Orponen and H. Mannila, On approximation preserving reductions: complete problems and robust measures, Tech. Report, University of Helsinki, 1987.

[13] C. Papadimitriou and K. Steiglitz, *Combinatorial Optimization: Algorithms and Complexity* (Prentice-Hall, Englewood Cliffs, NJ, 1982).

[14] C. Papadimitriou and M. Yannakakis, Optimization, approximation, and complexity classes, in: *Proc. 20th ACM Symp. Theory of Computing* (1988) 229–234.

[15] A. Paz and S. Moran, NP-optimization problems and their approximation, *Theoret. Comput. Sci.* **15** (1981) 251–277.

[16] S. Arora and S. Safra, Approximating clique is NP-complete, unpublished manuscript, January 1992.

[17] S. Arora, C. Lund, R. Motwani, M. Sudan and M. Szegedy, Proof verification and intractability of approximation problems, unpublished manuscript, May 1992.